UNISYS

CTOS®
Intelligent Data Communications Module System Software (IDMSS)
Administration Guide

Release 5.1

Priced Item

March 1992

Printed in U S America
4358 7260-000

UNISYS

CTOS®

Intelligent Data Communications Module System Software (IDMSS)

Administration Guide

Copyright © 1991 Unisys Corporation
All Rights Reserved
Unisys is a trademark of Unisys Corporation

Release 5.1

March 1992

Printed in USA

Priced Item

4358 7260–000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information or software material, including direct, indirect, special or consequential damages.

You should be careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

CTOS and SuperGen are registered trademarks of Convergent Technologies, Inc.

Context Manager and X-Bus are trademarks of Convergent Technologies, Inc.

BTOS is a trademark of Unisys Corporation.

Page Status

Page	Issue
iii	Original
iv	Blank
v through vii	Original
viii	Blank
ix through xii	Original
1–1 through 1–4	Original
2–1 through 2–9	Original
2–10	Blank
3–1 through 3–8	Original
4–1 through 4–6	Original
5–1 through 5–12	Original
Glossary–1 through 5	Original
Glossary–6	Blank
Index–1 through 4	Original

Contents

About This Guide ix

Section 1. Overview of Intelligent Modules

What Is an Intelligent Module? 1–1
Hardware Configurations 1–2
Software Components 1–2
Setting Up an Intelligent Module 1–3

Section 2. Installing IDMSS Services

Required System Services 2–1
Installing the XBIF Service 2–1
Installing IMSP 2–2
Modifying IMSP Parameter Values 2–2
IMSP Parameters 2–3
Installing During System Initialization 2–4
Modifying the System Initialization File 2–5
Specifying Parameter Values 2–6
Rebooting an Intelligent Module 2–7
Loadable Request Files 2–9

Section 3. Installing Communications Services

Methods 3–1
Using the IMRun Command 3–1
IMRun Command Parameters 3–3
IMRun Command Example 3–4
Customizing Executive Commands 3–5
Installing During System Initialization 3–7
Direct Installation Onto an Intelligent Module 3–8

Section 4. Customizing an Intelligent Module Operating System

Why Customize IMCTOS? 4–1
Interprocessor Request Blocks 4–1
User Numbers 4–2
What You Will Need to Customize IMCTOS 4–2
Modifying the Source Code 4–2
Editing the Prefix File 4–3
Customizable Parameters 4–3
Assembling and Linking 4–4
Booting a Customized Operating System 4–5
Booting Different Operating Systems on Multiple IDS and LAN Modules 4–5

Section 5. Troubleshooting

Error Reporting 5–1
IMSP Errors 5–1
IMRun Errors 5–1
Crash Routine 5–2
Messages in the System Log File 5–2
Dump Procedure 5–3
Running Out of Interprocessor Request Blocks 5–4
Error Codes 5–5
Intelligent Module Firmware 5–5
Intelligent Module Support Program (IMSP) 5–7
IMExecutive 5–9
Interprocess Communications (IPC) 5–10

Glossary

Index

Figures

1–1. IDMSS Software Components 1–3

2–1. Sample SysInit.jcl File 2–5

3–1. IMRun Command Used in SysInit.jcl File 3–7

Tables

2–1. IMBootstrap Parameter Fields 2–8

3–1. IMRun Parameter Fields 3–3

About This Guide

Who This Guide Is For

This manual is for system and network administrators who are responsible for installing and maintaining system software in a CTOS-based cluster.

What This Guide Covers

This manual provides procedures for using Intelligent Data Module System Software (IDMSS), version 5.1, on workstations equipped with one or more intelligent data communications modules. It also includes procedures for customizing the IDMSS package. See Section 1, "Overview of Intelligent Modules," for a description of IDMSS.

Changes to This Edition

This edition of the IDMSS manual has been completely rewritten. Changes in terminology and content are described below:

Terminology

- The name of the operating system for intelligent data communications modules, formerly IMBTOS, has been changed to IMCTOS.
- The software release documentation, formerly called Release Notes, Release Notice, or Release Information File, is now called the Software Release Announcement (SRA).

Content

- Information about hardware and software requirements has been moved to the SRA.
- Software installation procedures have been moved to the SRA.

- Customizable parameters for the Intelligent Module Support Program (IMSP) are now documented. See Section 2, "Installing IDMSS Services."
- Parameters and build procedures for customizing IMCTOS are now documented. See Section 4, "Customizing an Intelligent Module Operating System."

What Is New in the Product

The following changes and enhancements to IDMSS, version 5.1, are documented in this manual:

- Information about new intelligent-module hardware, including the Integrated Services Digital Network (ISDN) module and the Series 5000 Ethernet cartridge, is included throughout the manual.
- The intelligent module operating system (IMCTOS) can now load its own customized request file, *[Sys]<Sys>IDM>Request.sys*. See Section 2, "Installing IDMSS Services."
- If a workstation is equipped with more than one intelligent module of the same type, different versions of IMCTOS can be loaded onto each. See Section 4, "Customizing an Intelligent Module Operating System."
- IMCTOS creates its own crash dump file when necessary, during the crash dump procedure. See Section 5, "Troubleshooting."
- Message logging to the system log file has been improved. See Section 5, "Troubleshooting."

How This Guide Is Organized

This manual is divided into the following sections:

Section 1. Overview of Intelligent Modules

This section describes the types of intelligent modules, how they work, and the software components of IDMSS.

Section 2. Installing IDMSS Services

This section describes how to install the system services that are supplied with IDMSS and how to reboot an intelligent module.

Section 3. Installing Communications Services

This section describes several methods for installing communications system services on an intelligent module.

Section 4. Customizing an Intelligent Module Operating System

This section describes how to build a customized version of IMCTOS.

Section 5. Troubleshooting

This section describes how to identify problems and interpret error messages that are associated with IDMSS.

In addition, a glossary and index are included near the end of the manual.

Conventions

The following conventions are used throughout this manual:

- Command names are capitalized and appear in bold type, for example, **Install IMSP**.
- Names of fields appear in italics, for example, *Processor ID* field.
- Names of keys appear in small uppercase bold, for example, **GO**.
- File specifications appear in italics, for example, *[Sys]<Sys>SysInit.jcl*.

Where to Find More Information

For information about CTOS commands and system administration, see the following manuals:

CTOS Executive Reference Manual

CTOS System Administration Guide

For information about workstation hardware configurations, see the following manuals:

CTOS Workstations Planning and Owner's Maintenance Guide

SuperGen® Series 5000 Installation and Maintenance Guide

For information about network configurations, see the following manual:

CTOS Cluster and Network Hardware Installation Guide

For a detailed listing of error codes and their meanings, see the following manual:

CTOS Status Codes Reference Manual

Section 1
Overview of Intelligent Modules

What Is an Intelligent Module?

An intelligent module is a workstation component that contains its own central processing unit (CPU). Its purpose is to run communications software programs. Because it is a coprocessor with its own operating system, an intelligent module does not compete with the main CPU for processing time. Therefore, data communications operations receive top processing priority on the intelligent module, while the main CPU remains free for other processing tasks.

The following types of intelligent modules are available:

IDS	This module, sometimes called an Intelligent Data Slice (IDS or ID2), can run most data communications gateways and data links.
LAN	This group includes the Ethernet and Token-Ring modules and the Series 5000 Ethernet cartridge. They can run SNA transport or OSI transport local area network (LAN) software.
ISDN	This module can run the Integrated Services Digital Network (ISDN) software.

For a detailed list of supported hardware and software products, see the IDMSS Software Release Announcement.

Hardware Configurations

To be recognized by the workstation, intelligent modules must be properly positioned in relationship to other modules. In addition, for best performance, IDS modules should be located nearer to the main CPU than LAN or ISDN modules. See the *CTOS Workstations Planning and Owner's Maintenance Guide* for detailed information about hardware configurations.

A maximum of four intelligent modules can be recognized per workstation.

Software Components

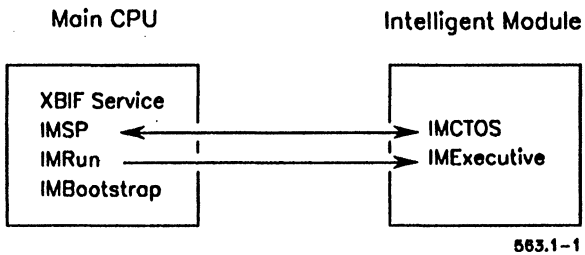
Intelligent Data Communications Module System Software (IDMSS) is a software package that makes an intelligent module a functional part of your workstation. IDMSS consists of the following system services and commands:

X-Bus™ Interface Service (XBIF)	This system service synchronizes the processing of intelligent modules with the main CPU.
Intelligent Module Support Program (IMSP)	This system service initializes the intelligent module and loads the operating system (IMCTOS) onto it. In addition, it provides interprocessor communications between the main CPU and the intelligent module.
IMCTOS	This operating system runs on the intelligent module. It is a real-mode, single-partition CTOS operating system without video, keyboard, or disk support. It is loaded by IMSP into memory on the intelligent module.

IMExecutive	This program is the command interpreter for the intelligent module. It is loaded into memory on the intelligent module.
IMRun command	This command is used to install system services on the intelligent module. It communicates with the IMExecutive on the intelligent module.
IMBootstrap command	This command is used to reboot an intelligent module.

Figure 1–1 illustrates the software components of IDMSS and how they communicate with one another.

Figure 1–1. IDMSS Software Components



Setting Up an Intelligent Module

Before you begin installation of the IDMSS software, the intelligent module should be in place on your workstation; see the appropriate hardware installation guide for your particular module.

In addition, the CTOS operating system and data communications software should be properly installed and configured. See the IDMSS Software Release Announcement for information about software interdependencies and other requirements.

To set up IDMSS and communications services, you will need to complete the following tasks:

1. Install the IDMSS software package onto the workstation. For installation instructions, see the Software Release Announcement.
2. Install the XBIF Service. See Section 2, "Installing IDMSS Services."
3. Install IMSP. See Section 2, "Installing IDMSS Services."
4. Install communications services on the intelligent module. See Section 3, "Installing Communications Services."

Section 2
Installing IDMSS Services

Required System Services

This section describes how to install the XBIF Service and the Intelligent Module Support Program (IMSP) into memory on the main CPU. They can be installed either with Executive commands or during system initialization. Procedures for both methods are described in this section.

The task of installing system services into processor memory is different from installing software applications onto a disk. If you are not familiar with CTOS system services, see the *CTOS System Administration Guide* for a detailed discussion.

Installing the XBIF Service

The XBIF Service synchronizes processing between the main CPU and intelligent X-Bus modules. Before you begin to install it, exit any Context Manager™ or Remote User Manager session that is running on the workstation. Then, follow these steps:

1. On the Executive command line, type **Install XBIF Service**.
2. Press GO.

You are informed when the installation is complete.

Note: *On workstations running 11.5 Standard Software, use the* **Install XBus Server** *command to install the XBIF Service.*

Installing IMSP

IMSP loads the intelligent module operating system (IMCTOS) and command interpreter (IMExecutive) into memory on the intelligent module. It then continues to run on the main CPU to provide interprocessor communications between the intelligent module and the main CPU.

Install the XBIF Service before you install IMSP. Then, to install IMSP, follow these steps:

1. On the Executive command line, type **Install IMSP**.
2. Press **GO**.

After IMSP is installed, a series of dots is displayed to indicate that IMCTOS is being loaded on the intelligent module. Multiple intelligent modules boot consecutively.

If an error message is displayed, see Section 5, "Troubleshooting." Nonrecoverable errors (those that prevent an intelligent module from booting) are logged in the system log file, which you can view or print with the **PLog** command. See the *CTOS Executive Reference Manual* for information about using **PLog**. IMSP attempts to boot an intelligent module twice before terminating.

Modifying IMSP Parameter Values

IMSP parameters have been set to values that are appropriate for many environments. However, depending on the communications software you plan to install on the intelligent module, you might need to change some of the IMSP parameters. In most cases, you need to modify parameters only when you build and install a customized version of IMCTOS on the intelligent module, as described in Section 4, "Customizing an Intelligent Module Operating System."

To modify parameter values, you install IMSP with the **Run** command; its run-file name is *[Sys]<Sys>IMSP.run*. See the *CTOS Executive Reference Manual* for information about using the **Run** command.

IMSP Parameters

When the parameters for IMSP are filled into the **Run** command form, they correspond to the *[Parameter n]* fields as numbered below. Do not change parameter values unless you have been instructed to do so by a technical support engineer or, if from past experience, you are certain that you need to do so.

[Parameter 1]

Back off timer. This field defines a value for the mode-3 DMA back off timer. Its value is set according to the module pointer on the X-Bus relative to other intelligent modules on the workstation. Do not change it unless you are instructed to do so by a Technical Support engineer.

[Parameter 2]

Number of RCBs. This field defines the number of Request Control Blocks (RCBs) to be allocated on the main CPU for outgoing requests. The default is 200.

[Parameter 3]

Number of large IPRBs. This field defines the number of large interprocessor request blocks (IPRBs) to be allocated on the main CPU for incoming requests. The default is 10.

[Parameter 4]

Size of large IPRBs. This field defines the size, in sectors, of large IPRBs. A sector is 512 bytes. The default is 5.

[Parameter 5]

Number of small IPRBs. This field defines the number of small IPRBs to be allocated on the main CPU for incoming requests. The default is 20.

[Parameter 6]

Size of small IPRBs. This field defines the size, in bytes, of small IPRBs. The default is 170.

[Parameter 7]

Watchdog interval. This field defines the interval, in tenths of a second, between watchdog requests. The watchdog checks the status of the intelligent module and reports crashes to the main CPU. The default is 50 (5 seconds).

[Parameter 8]

CycleWaitEmpty timer. This field defines a value for the CycleWaitEmpty timer. The default is 65535. (For a description of the CycleWaitEmpty timer, see "Messages in the System Log File," in Section 5, "Troubleshooting.")

[Parameter 9]

Number of user numbers. This field defines the number of user numbers to allocate. The default is 20.

[Parameter 10]

Number of pb links. This field defines the number of 6-byte structures to be allocated on the main CPU for storing the memory addresses of data items (pbs) with outstanding requests. The default is the number of RCBs multiplied by 6.

Installing During System Initialization

After you have successfully installed the XBIF Service and IMSP and are confident they are functioning correctly, you may want them to be installed automatically each time the system is rebooted.

To make this happen, you can include commands for them in *[Sys]<Sys>SysInit.jcl*, the system initialization JCL file. (If you are not familiar with the concept of system initialization and the *SysInit.jcl* file, see the *CTOS System Administration Guide.*)

A sample *SysInit.jcl* file is shown in Figure 2–1; entries for the XBIF Service and IMSP are shown in boldface type.

Figure 2–1. Sample SysInit.jcl File

```
$Job SysInit
$Run [Sys]<Sys>XBIF.run
$Run [Sys]<Sys>IMSP.run
$Run [Sys]<Sys>DLM.run
$Run [Sys]<Sys>Install802/3.run,35,,,T
$Run [Sys]<Sys>InstallTransNet.run
$Run [Sys]<Sys>STI.run,,,,2620
$Run [Sys]<Sys>Session.run,,2620,200
$Run [Sys]<Sys>BNet.run,ExNode,16,48,48,,10,,16,48,32
$Run [Sys]<Sys>NS.run
$End
```

Modifying the System Initialization File

To add commands to the system initialization JCL file, follow these steps:

1. On the Executive command line, type **Editor**; then press **RETURN**.
2. Type **[Sys]<Sys>SysInit.jcl**, as shown below:

   ```
   [File name(s)]  [Sys]<Sys>SysInit.jcl
   ```

3. Press **GO**.

   The file appears on the screen. It should look similar to the sample shown in Figure 2–1.

   Note: *If the file does not exist, you are informed; press* **GO** *again to create it. See the* CTOS System Administration Guide *for information about creating a system initialization JCL file from scratch.*

4. Add the **$Run** statements for XBIF and IMSP to the JCL file, as shown in bold type in Figure 2–1.
5. Press **FINISH**, then **GO**, to exit the Editor.

Note: *On systems running 12.0 Standard Software (including 3.3 CTOS II, 3.2 BTOS II, 2.4 CTOS/VM, and 3.0 CTOS/XE) and subsequent later versions, the dollar sign is not required in JCL files. It does not, however, interfere with execution of the file and is shown here for backward compatibility.*

Specifying Parameter Values

If you need to include parameter values for IMSP, specify them after the run file name, separated by commas, as shown in the following example:

```
$Run [Sys]<Sys>IMSP.run, 8, 200, 20, 5, 40, 170
```

Parameter values, as they follow the run file name, correspond to command form fields on a one-to-one basis in sequential order. If you need to leave a parameter value blank, enter a comma as a place holder. The following example shows the first, fourth, and fifth parameter values left blank:

```
$Run [Sys]<Sys>IMSP.run,,200, 10,,, 200
```

If a field requires subparameters (that is, more than one value per field), enclose subparameters in parentheses and separate them with commas, as shown in the following example:

```
$Run [Sys]<Sys>SNA.run,,,,B, 7,,,,,,, 9600,,,,,, (1,6)
```

See the *CTOS System Administration Guide* for more information about JCL file syntax.

Rebooting an Intelligent Module

Once an intelligent module is up and running, you can reboot it without rebooting the main CPU or reinstalling IMSP. You need to reboot an intelligent module to modify communications service parameters or to install a different communications service on it. In addition, you may need to reboot if a nonrecoverable error occurs.

To reboot an intelligent module, you use the **IMBootstrap** command, as described in the following steps:

1. On the Executive command line, type **IMBootstrap**; then press **RETURN**.
2. Fill in the command form, as shown in the following example. Parameter fields are described in Table 2–1.

```
IMBootstrap
 Processor ID              1
 File to bootstrap from    [Sys]<Sys>IDM>SysImage.sys
```

3. Press **GO**.

A message informs you that the intelligent module is booting and when booting is complete.

When an intelligent module is rebooted, all communications services are removed from its memory and must be reinstalled before they can be accessed. See Section 3, "Installing Communications Services."

Table 2–1. IMBootstrap Parameter Fields

Field Name	Description
Processor ID	Enter the processor ID of the intelligent module you want to boot. Intelligent module processor IDs are incremental from left to right, beginning with 1 for the first intelligent module.
File to bootstrap from	Enter the file specification for the intelligent module operating system you want to boot on the processor. File specifications for the prebuilt operating systems are listed below. See also "Booting a Customized Operating System," in Section 4. **IDS** *[Sys]<Sys>IDM>SysImage.sys, or* **LAN** *[Sys]<Sys>LAN>SysImage.sys* **ISDN** *[Sys]<Sys>ISDN>SysImage.sys*

Loadable Request Files

During bootstrapping, a request file is loaded into memory on the intelligent module. The request file contains supplements to IMCTOS, which are required to run particular system services or applications. In most cases, installation procedures for individual software products incorporate requests into the loadable request file when required.

The default system request file is named *[Sys]<Sys>Request.sys*. It contains supplemental request codes for all software on the workstation and is loaded by CTOS into memory on the main CPU. It is also the default request file loaded by IMCTOS into memory on an intelligent module.

Loadable requests, however, consume processor memory, and all requests contained in *Request.sys* are not needed on intelligent modules. Therefore, memory on an intelligent module can be conserved by loading only the request codes that are needed. For that to happen, request codes for the intelligent module must be placed into a special request file named *[Sys]<Sys>IDM>Request.sys*. If that file exists, it is loaded by IMCTOS instead of *[Sys]<Sys>Request.sys*.

[Sys]<Sys>IDM>Request.sys must include request codes for IDMSS, which are contained in a file named *Request.IDMSS.txt*, on the distribution media, and for the communications services you plan to run on the intelligent module. See the **List Request Set** and **Make Request Set** commands, in the *CTOS Executive Reference Manual*, for information about incorporating request codes into a loadable request file.

If *[Sys]<Sys>IDM>Request.sys* exists but the request codes you need are not merged into it, error 31 (no such request) occurs when you try to install communications services on the intelligent module.

Section 3
Installing Communications Services

Methods

This section describes the following methods for installing communications services on an intelligent module:

With the **IMRun** command	You can use this generic command to install most communications services onto an intelligent module.
With customized Executive commands	You can customize the command for a particular communications service by modifying your system command file.
During system initialization	Communications services can be installed during system initialization, after IMSP has booted the intelligent module.
Directly onto the intelligent module	Some communications services have been designed to be loaded directly onto the intelligent module. See the documentation that accompanies the product.

Using the IMRun Command

The **IMRun** command loads the communications-service run file you specify into memory on the intelligent module. It is similar to the Standard Software **Run** command, as described in the *CTOS Executive Reference Manual.*

To use the **IMRun** command, follow these steps:

1. On the Executive command line, type **IMRun**; then press **RETURN**.
2. Fill in the command form, as shown in the following example. Parameter fields are described in Table 3–1.

```
IMRun
 Processor ID     1
 Run file         [Sys]<Sys>X25PktInstall.run
 [Debug?]
 [Case]
 [Command]
 [Parameter 1]    [Sys]<Sys>X25.cfg
 [Parameter 2]
 [Parameter 3]
 [Parameter 4]
 [Parameter 5]
 [Parameter 6]
 [Parameter 7]
 [Parameter 8]
 [Parameter 9]
 [Parameter 10]
 [Parameter 11]
 [Parameter 12]
 [Parameter 13]
 [Parameter 14]
 [Parameter 15]
 [Parameter 16]
 [Parameter 17]
 [Parameter 18]
```

3. Press **GO**.

IMRun Command Parameters

Parameter fields for the **IMRun** command are described in Table 3–1.

Table 3–1. IMRun Parameter Fields

Field Name	Description
Processor ID	Enter the processor ID of the intelligent module on which you want to install the system service. Intelligent module processor IDs are incremental from left to right, beginning with 1 for the first intelligent module.
Run file	Enter the file specification of the system service you want to install. The format of the file specification is *{Node}[Volume]<Dir>FileName^pswd.* The file name is required; if the node, volume, directory are not specified, default values are taken from the current path setting.
[Debug?]	This parameter is not implemented.
[Case]	Default: 00 Enter a two-character case value. Case values are intrinsic to the run file and identify individual functions if the run file is capable of executing more than one function.
[Command]	Default: None Enter the command name you want to use to execute the run file.
[Parameter 1] to *[Parameter 18]*	Default: None Enter parameters in the appropriate command form field. Parameters correspond on a one-to-one basis with the command form for the system service you are installing. See the **IMRun** command example, later in this section.

IMRun Command Example

The following example shows how to fill in the **IMRun** command form to install the SNA Transport Service (run-file name *SNATS.run*) on an IDS module, which is placed as the first intelligent module on the system. For further clarification, the corresponding command form for the **Install SNA Transport Service** command is also shown. Note that parameter values, beginning with *[Parameter 1]* of the **IMRun** command form, correspond on a one-to-one basis with the parameter fields of the **Install SNA Transport Service** command form.

IMRun command form

```
IMRun
 Processor ID     1
 Run file         [Sys]<Sys>SNATS.run
 [Debug?]
 [Case]
 [Command]
 [Parameter 1]    SNADLL.cfg                    [SDF name]
 [Parameter 2]
 [Parameter 3]
 [Parameter 4]    530                      [Max PIU size]
 [Parameter 5]
 [Parameter 6]
 [Parameter 7]
 [Parameter 8]
 [Parameter 9]
 [Parameter 10]
 [Parameter 11]
 [Parameter 12]   20                 [Max # indep sessions]
 [Parameter 13]
 [Parameter 14]
 [Parameter 15]
 [Parameter 16]
 [Parameter 17]
 [Parameter 18]
```

Corresponding Install SNA Transport Service command form

```
Install SNA Transport Service
 [SDF name]                          SNA.cfg        (Param 1)
 [DLM node name]                     _____________
 [Node type]                         _____________
 [Max PIU size]                      530            (Param 4)
 [LUs 3270p]                         _____________
 [    3270 ]                         _____________
 [    RJE  ]                         _____________
 [    APPC ]                         _____________
 [    LUIS ]                         _____________
 [ServiceID]                         _____________
 [Trace buffer size]                 _____________
 [Max # indep sessions]              20             (Param 12)
 [Max # net man applications]        _____________
 [AutoDisconnect?]                   _____________
```

Customizing Executive Commands

Another method for installing communications services on an intelligent module is to customize the Executive command supplied with the communications software package so that it installs on the intelligent module rather than the main CPU.

Executive commands are usually contained in a command file named *[Sys]<Sys>Sys.cmds*. However, special-purpose command files can be assigned on a user-name basis, and that could be the case on your workstation. See the *CTOS Executive Reference Manual* for detailed information about command files and the *CTOS System Administration Guide* for information about assigning command files to specific user names.

A command file contains the command form, run-file name, and other information about each command. It can be modified with the **Command File Editor** command, which is supplied with Standard Software.

To customize a command so that it installs the communications service onto an intelligent module, you need to modify the command that is supplied with the communications software package, as described in the following steps:

1. On the Executive command line, type **Command File Editor**; then press **RETURN**.
2. In the command form, type **[Sys]<Sys>Sys.cmds** (or the appropriate command file name), as shown below:

```
Command File Editor
  [Command file]  [Sys]<Sys>Sys.cmds
```

3. Press **GO**.
4. Position the highlight on the name of the command you want to modify, for example, **Install SNA Transport Service**.
5. Press **F1** (Edit).
6. Position the highlight on the *Run file* field near the bottom on the screen.
7. Type the processor ID number, surrounded by colons (:), in front of the run file specification that appears in the field, for example:

   :1:*[Sys]<Sys>SNATS.run*

8. Press **GO**.
9. Press **FINISH**, then **GO**, to exit and save the file.

See the *CTOS Executive Reference Manual* for more detailed information about the **Command File Editor** command.

CAUTION

The **IMRun** command itself cannot be edited with the **Command File Editor**. Do not attempt to select or alter **IMRun**.

Installing During System Initialization

You can install communications services on an intelligent module during system initialization. To do so, you use the **$Run** statement and the run file name for the **IMRun** command (*[Sys]<Sys>IMRun.run*), followed by the run file name and parameter values for the communications service you want to install, in the system initialization JCL file.

A sample *SysInit.jcl* file is shown in Figure 3–1; **IMRun** command entries are shown in boldface type.

See "Installing During System Initialization," in Section 2, for an overview of system initialization functions. See Table 3–1 for a description of **IMRun** parameters. See the *CTOS System Administration Guide*, if you are not familiar with the concept of system initialization.

Figure 3–1. IMRun Command in a System Initialization JCL File

```
$Job SysInit
$Run [Sys]<Sys>XBIF.run
$Run [Sys]<Sys>IMSP.run
$Run [Sys]<Sys>BNet.run,Pubs,8,32,32,,15,,8,32,8
$Run [Sys]<Sys>DLM.run
$Run [Sys]<Sys>IMRun.run,1,[Sys]<Sys>SDLC.run
$Run [Sys]<Sys>IMRun.run,1,[Sys]<Sys>SNATS.run,[Sys]<Sys>SNA.cfg,,,530,,,,,,,20
$Command Install Mouse Service
$Command Install Mail Service
$Run [Sys]<GPS>GpsInstall.run
$End
```

To add communications services to the system initialization JCL file, follow these steps:

1. On the Executive command line, type **Editor**; then press **RETURN**.
2. Type **[Sys]<Sys>SysInit.jcl**, as shown below:

```
[File name(s)]  [Sys]<Sys>SysInit.jcl
```

3. Press **GO**.

   The file appears on the screen. It should look similar to the sample shown in Figure 3–1.

   Note: *If the file does not exist, you are informed; press* **GO** *again to create it. See the* CTOS System Administration Guide *for information about creating a system initialization JCL file from scratch.*

4. Add the **$Run** statements for the communications services you want to install. For an example, see Figure 3–1.
5. Press **FINISH**, then **GO**, to exit the Editor.

Direct Installation Onto an Intelligent Module

Some communications services are designed to load directly onto the intelligent module. For example, with the **Install Token Ring Data Link Layer** command, you can install that system service directly on an intelligent token-ring LAN module.

When such a command exists for a communications service, you do not use the **IMRun** command to install it. Simply execute the command as instructed in the manual for the product.

Similarly, you do not use **IMRun** parameters to install such system services during system initialization. Simply specify the run-file name and parameter values for the communications service after the **$Run** statement in the *SysInit.jcl* file. See the *CTOS System Administration Guide* for detailed information about adding entries to the system intialization JCL file.

Section 4
Customizing an Intelligent Module Operating System

Why Customize IMCTOS?

This section describes how to build (also called *SysGen*) an IMCTOS operating system with customized parameters. You would do this when a communications service in your environment requires more interprocessor request blocks (IPRBs) or user numbers than those provided in the released version of IMCTOS.

IPRBs and user numbers are customizable parameters in IMCTOS. Their relationship to communications applications running on an intelligent module are described below.

Interprocessor Request Blocks

Communications between the main CPU and the intelligent module are handled in a manner similar to cluster communications. Request blocks are carried across the X-Bus in data structures called interprocessor request blocks (IPRBs). (See the *CTOS Operating System Concepts Manual* for detailed information about request blocks.) IMCTOS defines two sizes of IPRBs to accommodate large and small request blocks. Those sizes are preset to optimal values for most intelligent module applications.

In addition to the size of IPRBs, IMCTOS defines the number of large and small IPRBs to be allocated in memory on the intelligent module; those are the values you might need to customize. The number of IPRBs defines the maximum number of outstanding requests that can be served by IMCTOS. To determine the number of IPRBs your system needs, find out how many outstanding request blocks can be served by your communications product (see the documentation for the communications product, possibly the SRA). See also Section 5, "Troubleshooting," for information about running out of IPRBs.

User Numbers

Application partitions that run on CTOS workstations are assigned unique user numbers. IMCTOS maintains a table of user numbers that mirrors those assigned on the main CPU; the size of that table is a value you might need to customize.

If the IMCTOS user-number table becomes full, error 31756 occurs. See the *CTOS Operating System Concepts Manual* for more detailed information about user numbers.

What You Will Need to Customize IMCTOS

The utilities you use to customize IMCTOS are called the IMCTOS Build Utilities; they are included with the IDMSS software as a separately installable package. See the SRA for IDMSS for installation instructions.

In addition to the IMCTOS Build Utilities, you will need to install the CTOS Development Utilities on the workstation where you plan to build the customized operating system. See the SRA for the Development Utilities for installation instructions.

Modifying the Source Code

Customizable source code is contained in prefix files, which you modify to customize IMCTOS. The prefix files are named as follows:

IDS modules

[Sys]<IMCTOSGen>IDM>ImCtosSp.asm

LAN modules and cartridges

[Sys]<IMCTOSGen>LAN>ImCtosSp.asm

ISDN modules

[Sys]<IMCTOSGen>ISDN>ImCtosSp.asm

Editing the Prefix File

To edit a prefix file, follow these steps:

1. Use the **Path** command to set your path to the build directory for IMCTOS.
2. On the Executive command line, type **Editor**, then press **RETURN**.
3. Type the file name of the prefix file you want to modify, as shown in the following example:

```
Editor
 File name(s)              IDM>ImCtosSp.asm
 [Read only?]
 [Alternate user name]
```

4. Press **GO** to open the file.
5. Edit the parameter(s) you want to customize; see "Customizable Parameters," below.
6. After you have finished making changes, press **FINISH**, then **GO**, to exit the Editor and save the file.

Customizable Parameters

nIPRBLarge

Defaults: IDM, 20; ISDN, 20; LAN, 15

This parameter defines the number of large IPRBs to be allocated on the intelligent module.

nSectorIPRBLarge

Default: 5

This parameter defines the size, in sectors, of large IPRBs. A sector is 512 bytes.

nIPRBSmall

Default: IDM, 20; ISDN, 60; LAN, 100

This parameter defines the number of small IPRBs to be allocated on the intelligent module.

sIPRBSmall

Default: 170

This parameter defines the size, in bytes, of small IPRBs.

nUserNumMax

Default: 500

This parameter defines the maximum number of user numbers that can be accommodated in the IMCTOS user-number table. Each user number requires 8 bytes.

IPRBQueueTimer

Default: 10

This parameter defines the interval, in tenths of a second, at which messages about running out of IPRBs are written to the system log file. (See Section 5, "Troubleshooting," for information about running out of IPRBs how that is noted in the system log file.) The default is 10 (one second). In cases where IPRB messages are being logged constantly, however, this interval increases logarithmically, to prevent the system log file from filling with repetitions of the same message. The maximum interval between messages is one hour. To suppress the message, set the value of this parameter to 0.

Assembling and Linking

After you have made changes to the prefix file, you assemble and link it to build a new operating system, as described in the following steps:

1. Use the **Path** command to set your path to the build directory for IMCTOS.
2. On the Executive command line, type **Assemble**; then press **RETURN**.
3. Enter the name of the prefix file, as shown in the following example:

```
Assemble
 File name        IDM>ImCtosSp.asm
 [Errors only?]
```

4. Press **GO**.

   If no error messages occur during the assembly operation, proceed to the next step. If errors occur, see "Troubleshooting SysGen Errors," in the *CTOS System Administration Guide*.

5. On the Executive command line, type **Link IMCTOS**; then press **RETURN**.

6. Enter the intelligent module type (IDM, ISDN, or LAN) and version number, as shown in the following example:

```
Link IMCTOS
 Operating system type  IDM
 Version                5.1-custom
```

   The date and time are automatically appended to the version number.

7. Press **GO**.

   If errors occur during linking, see "Troubleshooting SysGen Errors," in the *CTOS System Administration Guide*.

Booting a Customized Operating System

The System Image file for the new operating system is created in the build directory; it does not overwrite the System Image file for the current operating system in *[Sys]<Sys>*. Use the **IMBootstrap** command to test the new operating system before you overwrite the System Image file for the intelligent module in *[Sys]<Sys>*.

For the customized operating system to be booted automatically by IMSP, it must be copied to the appropriate System Image file, as listed below:

- *[Sys]<Sys>IDM>SysImage.sys* for IDS modules,
- *[Sys]<Sys>LAN>SysImage.sys* for LAN modules and cartridges, or
- *[Sys]<Sys>ISDN>SysImage.sys* for ISDN modules.

Booting Different Operating Systems on Multiple IDS and LAN Modules

If the workstation is equipped with more than one IDS module, each can boot a different operating system.

To boot a customized operating system on the first IDS module, copy it to the following System Image file:

[Sys]<Sys>IDM1>SysImage.sys

To boot a customized operating system on the second IDS module, copy it to the following System Image file:

[Sys]<Sys>IDM2>SysImage.sys

If either file does not exist, IMSP boots *[Sys]<Sys>IDM>SysImage.sys*, the default System Image file for IDS modules.

The same is true for LAN modules and cartridges. That is, copy the customized operating system to *[Sys]<Sys>LAN1>SysImage.sys* for the first LAN device, and to *[Sys]<Sys>LAN2>SysImage.sys* for the second LAN device. If either file does not exist, *[Sys]<Sys>LAN>SysImage.sys* is the default System Image file.

Section 5
Troubleshooting

Error Reporting

This section describes errors related to the IDMSS components, including IMSP, interprocess communications (IPC), IMExecutive, and their associated commands. Errors related to communications applications are handled as if they occurred on the main CPU. See the *CTOS Status Codes Reference Manual* or the documentation for the communications product for information about application errors.

IMSP Errors

When an error occurs during the execution of **Install IMSP** or **IMBootstrap**, a numeric error code and a message are displayed. In addition, a description of the error is written to the system error log, which you can view or print with the **PLog** command (see the *CTOS Executive Reference Manual*). When IMSP installation is successful, Error 0 and a message are recorded in the system log file. See "Error Codes," later in this section, for a description of IMSP errors.

If you press ACTION-FINISH to terminate the **Install IMSP** command, the next time you execute **Install IMSP**, you are informed that the intelligent module was left in an unknown state. The intelligent module is then reinitialized and another attempt is made to install IMSP.

IMRun Errors

When you execute the **IMRun** command, the *Processor ID* parameter is checked. It must be a nonzero, one-digit number. If it is not, an error message is displayed.

ACTION-FINISH does not terminate the **IMRun** command. Therefore, pressing it does not terminate installation of a communications service. If you need to reinstall the communications service, execute **IMBootstrap** before reexecuting **IMRun**.

Crash Routine

When IMSP detects a nonrecoverable error on the intelligent module, the following events take place:

1. A message is written to the system log file.
2. A crash dump of the intelligent module is performed.
3. The intelligent module is rebooted.

System log file messages and the dump procedure are described in more detail below.

Messages in the System Log File

When IMSP detects a nonrecoverable error, a message is written to the system log file. The system log file can be viewed with the **PLog** command, as described in the *CTOS Executive Reference Manual*. Log file messages for IMSP and IMCTOS are described below.

Unknown crash detected by IMSP

This message is logged, along with the specific hardware register numbers, when IMSP detects a crash of the intelligent module. If the problem persists, contact your technical support representative for interpretation of the error.

Watchdog timer expired

IMSP periodically issues a watchdog request to IMCTOS to verify that it is running. If IMCTOS does not respond within the specified watchdog interval, IMSP detects a crash and reboots the intelligent module.

The lack of response can be caused by an application process running at a higher priority than IMCTOS, by an application that disables interrupts for an excessively long period, or by a problem with IMCTOS itself.

If the problem persists, try doubling or tripling the watchdog interval (parameter 7) in IMSP. See "Modifying IMSP Parameter Values," in Section 2, for information about setting IMSP parameters.

CycleWaitEmpty timer expired

When IMSP sends a request to IMCTOS, execution of intelligent-module processor instructions is momentarily interrupted. This is normal; however, if the intelligent module does not acknowledge the interrupt within a specified interval, IMSP detects a crash and reboots the intelligent module.

The lack of response can be caused by an application that disables interrupts for an excessively long period, by an application that has left interrupts disabled, or by IMCTOS crashing.

If the problem persists, try doubling the value of the CycleWaitEmpty timer (parameter 8) in IMSP. See "Modifying IMSP Parameter Values," in Section 2, for information about setting IMSP parameters.

Crash detected by IMCTOS

This error occurs when IMCTOS detects a fatal error and notifies IMSP that it can no longer continue. IMSP then initiates a crash dump and reboots the intelligent module.

Additionally, a crash status message may be written to the system log file. See the *CTOS Status Codes Reference Manual* for information about interpreting processor crashes.

Dump Procedure

The crash routine copies an image of system memory to a crash dump file. That file can then be analyzed by an operating system engineer to determine the cause of the crash. The crash dump file names for intelligent modules are listed below:

IDS

[Sys]<Sys>IDM>CrashDump.sys

LAN

[Sys]<Sys>LAN>CrashDump.sys

ISDN

[Sys]<Sys>ISDN>CrashDump.sys

If the appropriately named crash dump file does not exist, it is created during the crash dump procedure. If adequate disk space is not available for the crash dump file, a message is written to the system log file and the crash dump does not take place.

Running Out of Interprocessor Request Blocks

When an intelligent module runs out of interprocessor request blocks (IPRBs) the following symptoms may occur:

- Poor performance of communications services that are running on the intelligent module
- Poor performance of the cluster or of the entire network to which the intelligent module is connected
- Inability of the intelligent module to continue its processing activity

To assist you in diagnosing such problems, IMCTOS writes a message to the system log file when IPRBs are not available to serve requests. Such a message reads as follows:

Warning – IPRB exhaustion on processor *X*, pending queue depth *Y*.

where

X	Is the processor number of the intelligent module, and
Y	Is the number of requests currently waiting for an IPRB.

If you find that IPRBs are constantly being exhausted, you can build a customized version of IMCTOS that allocates more IPRBs on the intelligent module. See Section 4, "Customizing an Intelligent Module Operating System," for more detailed information.

In addition, you can change the interval at which the IPRB message is written to the system log file. See the *IPRBQueueTimer* parameter in Section 4.

Error Codes

Error codes for IDMSS are listed below. When displayed, each error message is preceded by the processor ID for the intelligent module on which it occurred.

Intelligent Module Firmware

31600	Internal error. Contact your technical support representative.
31601 to 31624	Intelligent Module hardware fault. A hardware fault has occurred on the intelligent module. IMSP will reboot the module automatically. If the error occurs twice in a row, IMSP will terminate and not reboot the module. In that case, the workstation should be rebooted and the IMSP reinstalled. If the error still occurs, run diagnostics and consult a customer service engineer.
31625	Stray software NMI. The intelligent module received a stray software nonmaskable interrupt. A user program may be sending the module interrupts. IMSP will reboot the module automatically.
31626 to 31629	Same as 31601 to 31624.
31630	IMSP structure not found. IMSP software structures are being corrupted. A user program may be overwriting these locations. IMSP will reboot the module automatically.

31631	Invalid command in IMSP structure. IMSP software structures are being corrupted. A user program may be overwriting these locations. IMSP will reboot the module automatically.
31632	Invalid destination address in IMSP structure. IMSP software structures are being corrupted. A user program may be overwriting these locations. IMSP will reboot the module automatically.
31633	Invalid source address in IMSP structure. IMSP software structures are being corrupted. A user program may be overwriting these locations. IMSP will reboot the module automatically.
31634	Invalid start address in IMSP structure. IMSP software structures are being corrupted. A user program may be overwriting these locations. IMSP will reboot the module automatically.
31635	Bad checksum. The intelligent module System Image has an invalid checksum. The copy of this file on the hard disk is probably corrupted. Reinstall the file from the installation media; then reinstall IMSP.
31636	Bad LAN address PROM checksum. The address PROM on the intelligent module has a bad checksum. This may indicate a hardware failure. Hardware diagnostics should be performed by a customer service engineer.

31637	Bad LAN address PROM contents. The address PROM on the intelligent module contains invalid data. This may indicate a hardware failure. Hardware diagnostics should be performed by a customer service engineer.
31638 to 31699	Internal error. Contact your technical support representative.

Intelligent Module Support Program (IMSP)

31700	Bad run file signature. The intelligent module System Image file has a bad signature in the run file header. The copy of this file on the hard disk is probably corrupt. Reinstall the file from the installation media; then reinstall IMSP.
31701	No intelligent module on the current configuration. IMSP can only be installed on a system that has an intelligent module. If a module is on the configuration and this error occurs, it may be due to a faulty connection. Turn off the workstation and reconnect the intelligent module. If the error still occurs, it may be a module hardware problem. Run diagnostics and consult a customer service engineer.
31702	Software timed out. This error occurs when IMSP attempts to reboot an intelligent module but does not receive a response. The "software timed out" message refers to IMSP abandoning its attempts to reboot the intelligent module. In many cases, this error is the result of a hardware problem on the intelligent module. In such a case, the intelligent module must be replaced or repaired to correct the error.

	In other cases, this error occurs when the application running on the intelligent module crashes and leaves memory in a state that cannot be dumped. In such a case, the 31702 error indicates that the module cannot be re-booted by IMSP. Such errors can sometimes be corrected by rebooting the workstation.
31703	Reserved for future use.
31704	Intelligent module is in an unknown state. The IDMSS software is not communicating properly. A user program may be sending the module spurious interrupts. IMSP will reboot the module automatically.
31705	Invalid values found in run file header. The System Image file has invalid values in the run file header. The copy of this file on the hard disk is probably corrupt. Reinstall the file from the installation media; then reinstall IMSP.
31706	Invalid processor number. An internal software error has occurred. IMSP will reboot the module automatically.
31707	Invalid workstation type. IMSP cannot be installed on an unsupported workstation. See the Software Release Announcement.
31708	Could not access XBIS. IMSP was not granted access to the XBIS, a CTOS structure that controls initialization of modules. A user program may not be following the proper protocol when using the XBIS. IMSP cannot be installed while such a program is running.

31709	Invalid operation on module. An internal software error has occurred. IMSP will reboot the module automatically.
31710 to 31729	Reserved for future use.

IMExecutive

31730	No run file specification. This is used for communication among the IMExecutive system service, the IMExecutive, and the IMRun utility. It does not indicate an error.
31731	No termination status. This is used for communication among the IMExecutive system service, the IMExecutive, and the IMRun utility. It does not indicate an error.
31732	A software error has occurred causing the IMExecutive system service to receive both an IMRun request, asking for status without presenting a run file specification, and an IMStatus request, asking for a run file specification, without presenting a termination status.
31733	A software error has occurred, causing a duplicate IMRun request to be received.
31734	A software error has occurred, causing a duplicate IMStatus request to be received.
31735	A software error has occurred, causing an unexpected request to be received by the IMExecutive system service.
31736	A program has exited with a termination status message that is too large for the IMExecutive system service to handle.

31737	An attempt has been made to pass a run file specification that is too large for the IMExecutive system service to handle. This indicates a software error because the IMExecutive will handle the largest legal file specification.
31738	An attempt has been made to pass a password that is too large for the IMExecutive system service to handle. This indicates a software error because the IMExecutive will handle the largest legal password.
31739	An attempt has been made to pass a Variable Length Parameter Block (VLPB) that is too large for the IMExecutive to handle.
31740	IMRun has been supplied with a Variable Length Parameter Block (VLPB) containing too many parameters.
31741	A software error has occurred, causing an internal consistency check to fail. The IMExecutive system service has received a swapping request for a partition that appears to be already swapped.
31742 to 31749	Internal error. Contact your technical support representative.

Interprocess Communications (IPC)

31750	A request has been received by IPC with an invalid processor ID. Examine the system log file to ensure that each intelligent module has been recognized and has bootstrapped. If so, a nonexistent processor has been specified by a software error.

31751	A request destined for a currently inactive processor has been received by IPC. The processor may have failed to boot initially or may have failed during operation. Examine the system log file to determine the cause of the failure.
31752	When IPC issues requests for abnormal termination, this error code is placed in the erc field of each request block. This error code is received only by system services and is not for operator display.
31753	A software or hardware error has occurred, causing an interrupt to be received from an inactive processor.
31754	A software or hardware error has occurred, causing an illegal value to be encountered in the type code field of a processor's mailbox.
31755	An attempt has been made to send via IPC a request that is too large for Interprocessor Request Blocks (IPRBs) of the destination processor.
31756	A processor has received requests from too many users, that is, different nodes, workstations, or partitions.
31757	Sufficient currently outstanding requests have been issued from a processor to consume all of the Request Control Blocks (RCBs).
31758	Sufficient currently outstanding requests have been issued to a processor to consume all of the Interprocessor Request Blocks (IPRBs).
31759	An attempt has been made to reboot a processor that is currently inactive and is in the process of rebooting.
31760	A software or hardware error has occurred, causing the IPC to receive a duplicate reboot notification request.

31761	A software or hardware error has occurred, causing IPC to receive an interrupt during initialization.
31762	A software or hardware error has occurred, causing an unexpected request to be received by the request manager.

Glossary

B

bootstrap (*also* boot)

The process of loading an operating system into memory on a processor.

C

central processing unit (CPU)

See processor.

command file

A file the Executive reads to display command forms.

communications service

A system service that handles requests for a communications application. *See also* system service.

coprocessor

A supplemental processor, in addition to the main CPU, that runs its own operating system. *See also* main CPU.

CPU

Acronym for central processing unit. *See* processor.

crash dump

The process of writing data from system memory into a file, so that it can be analyzed and debugged.

CTOS

The operating system used on CTOS workstations.

E

Ethernet module (or cartridge)

A LAN module that provides access to an Ethernet network.

I

IDMSS

See Intelligent Data Communications Module Systems Software.

IDS

A intelligent module coprocessor that specializes in the execution of data communication software.

IMBootstrap command

A utility that reboots an intelligent module.

IMCTOS

The intelligent module operating system.

IMExecutive

The IMCTOS command interpreter.

IMRun command

A utility that interfaces with the IMExecutive to execute programs on the intelligent module.

IMSP

See Intelligent Module Support Program.

Intelligent Data Communications Module Systems Software (IDMSS)

A software package that makes an intelligent communications module a functional part of a CTOS workstation.

intelligent module

A workstation module (or Series 5000 cartridge) that contains its own processing unit and runs its own operating system.

Intelligent Module Support Program (IMSP)

A program that initializes intelligent modules and loads their operating systems and other required files. IMSP also runs as a system service on the main CPU to provide interprocessor communications between it and the intelligent module.

interprocess communications (IPC)

The means by which a client process and a system service communicate with one another.

ISDN module

An intelligent module that provides access to an Integrated Services Digital Network (ISDN).

J

Job Control Language (JCL)

The programming language used in the system initialization file (*SysInit.jcl*) to automatically install system services each time the workstation is booted.

L

LAN module

An intelligent module used to create a local area network environment.

Local Area Network (LAN)

A data communication network of multiple nodes using peer-to-peer protocol over moderate distances.

M

main CPU

The central processing unit within the processor module that controls the workstation.

module

A workstation component that is housed in its own casing and connected as an individual unit to the workstation.

O

operating system

The program that controls execution of all other programs on a processor.

OSI Transport LAN

The software that runs with IDMSS on a LAN module to provide a local area network environment.

P

processor

The computer component that interprets and executes instructions.

R

request file

A binary file containing request definitions for a system service.

S

Software Release Announcement (SRA)

A version-specific document containing information about a new release of a software product. SRAs were formerly called Release Notes, Release Notices, or Release Information Files.

Standard Software

A set of basic commands and utilities that are packaged with the operating system.

SysGen

The process of building a customized operating system.

system error log

A file containing information about hardware and software errors. It can be displayed or printed with the **PLog** command.

system service

A program that expands the capabilities of the operating system.

T

Token Ring

A network with a ring topology that uses a token as the control device.

Token Ring module

A LAN module that provides access to a Token Ring network.

X

X-Bus Interface Service (XBIF)

A system service that synchronizes main CPU, LAN, and IDS processing.

Index

A

Assemble command, 4-4

B

Booting, intelligent module
  IMBootstrap command, 2-7
  System Image file names, 2-8
Building. *See* customizing IMCTOS.

C

Command File Editor command, 3-5
Command files
  customizing, 3-6
  special-purpose, 3-5
  Sys.cmds, 3-5
Commands, by name
  Assemble, 4-4
  Command File Editor, 3-5
  Editor, 2-5, 3-8, 4-3
  IMBootstrap, 1-3, 2-7, 5-1
  IMRun, 1-3, 3-1, 5-1
  Install IMSP, 2-2, 5-1
  Install XBIF Service, 2-1
  Install XBus Server, 2-1
  Link IMCTOS, 4-5
  List Reqeust Set, 2-9
  Make Reqeust Set, 2-9
  PLog, 2-2, 5-1
  Run, 2-2
Communications services. *See* installing communications services.
Context Manager, 2-1
Crash dumps, 5-2, 5-3
Crashes, intelligent module
  dump files, 5-3
  routines, 5-2
Customizing IMCTOS
  Assemble command, 4-4
  booting, 4-5
  build procedures, 4-4
  IPRBs, 4-1
  Link IMCTOS command, 4-5
  parameters, 4-3
  prefix files, 4-2
  software requirements, 4-2
  System Image file names, 4-5
  user numbers, 4-2
CycleWaitEmpty timer, 2-4, 5-3

D

Development Utilities, 4-2
Dumps, 5-2, 5-3

E

Editor command, 2-5, 3-8, 4-3
Error 31 (no such request), 2-9
Error codes
  31 (no such request), 2-9
  IMExecutive, 5-9
  IMSP, 5-7

Error codes *(cont.)*
  intelligent module firmware, 5-5
  IPC, 5-10
Error reporting
  IMRun command, 5-1
  IMSP, 5-1
Errors, nonrecoverable, 5-2
Ethernet module, 1-1
Executive
  command files, 3-5
  customizing commands, 3-5
  IMExecutive, 1-3
  Sys.cmds, 3-5

F

Fatal error. *See* errors, nonrecoverable.

H

Hardware configurations, 1-2

I

IDM>ImCtosSp.asm, 4-2
IDM>Request.sys, 2-9
IDS and ID2 modules, 1-1
IMBootstrap command
  command form, 2-7
  customized operating system, 4-5
  description, 1-3
  errors, 5-1
  parameter fields, 2-8
  System Image file names, 2-8
IMCTOS
  booting, 4-5
  customizing (SysGen), 4-1. *See also* customizing IMCTOS.
  description, 1-2
  prefix files, 4-2
  System Image file names, 4-5
IMExecutive
  description, 1-3
  error codes, 5-9
IMRun command
  command form, 3-2
  description, 1-3, 3-1
  errors, 5-1
  example, 3-4
  parameter fields, 3-3
IMSP. *See* Intelligent Module Support Program.
Install IMSP command
  errors, 5-1
  parameters, 2-3
Install XBIF Service command, 2-1
Install XBus Server command, 2-1
Installing communications services
  customized Executive commands, 3-5
  direct installation, 3-8
  IMRun command, 3-1
  methods, 3-1
  system initialization, 3-7
Integrated Services Digital Network (ISDN) module, 1-1
Intelligent Data Slice (IDS or ID2) module, 1-1
Intelligent module
  booting, 4-6
  communications services, installing. *See* installing communications services.
  crash dumps, 5-3
  crash routines, 5-2

Intelligent module *(cont.)*
customizing Executive commands for, 3-5
firmware error codes, 5-5
IMBootstrap command, 2-7
operating system, 1-2
processor ID, 2-8, 3-3
rebooting, 2-7
System Image file names, 2-8
Intelligent Module Support Program (IMSP)
bootstrap errors, 2-2
CycleWaitEmpty timer, 2-4
error codes, 5-7
Install IMSP command, 2-2
installing, 2-2, 2-4
IPRBs, 2-3
pb links, 2-4
user numbers, 2-4
watchdog, 2-4
Interprocessor request blocks (IPRBs)
IMCTOS, 4-1
IMSP, 2-3
queue timer, 4-4
running out of, 5-4
system log file, 5-4
IPC error codes, 5-10
IPRBs. *See* interprocessor request blocks.
ISDN module, 1-1
ISDN>ImCtosSp.asm, 4-2

L

LAN module and cartridge, 1-1
LAN>ImCtosSp.asm, 4-2
Link IMCTOS command, 4-5
List Request Set command, 2-9
Loadable request files
IDM>Request.sys, 2-9
Request.IDMSS.txt, 2-9
Request.sys, 2-9
Local Area Network (LAN) module and cartridge, 1-1
Log file, system. *See* system log file.

M

Make Request Set command, 2-9

N

No such request, (error 31), 2-9

P

PLog command, 2-2, 5-1
Prefix files, 4-2
Processor crashes, 5-3
Processor ID, 2-8, 3-3

R

RCBs, 2-3
Remote User Manager, 2-1
Request Control Blocks (RCBs), 2-3
Request files, 2-9
Request.IDMSS.txt, 2-9
Request.sys, 2-9
Run command, 2-2

S

Setup, 1-3
Software components
IMBootstrap command, 1-3
IMCTOS, 1-2

Software components *(cont.)*
IMExecutive, 1-3
IMRun command, 1-3
Intelligent Module Support Program (IMSP), 1-2
XBIF Service, 1-2
Status codes. *See* error codes.
Sys.cmds, 3-5
SysGen, 4-1
SysInit.jcl file
modifying, 2-5
sample of, 2-5, 3-7
syntax of, 2-6
System Image file names, 2-8, 4-5
System initialization
installing communications services, 3-7
installing IDMSS services, 2-4
SysInit.jcl file, 2-4 to 2-6, 3-7
System log file
IPRBs, 5-4
nonrecoverable errors, 5-2
System services, installing, 2-1

T

Timers
CycleWaitEmpty, 2-4, 5-3
watchdog, 2-4, 5-2
Token-Ring module, 1-1
Troubleshooting
crashes, 5-2
dump files, 5-3
error codes, 5-5
error reporting, 5-1

U

User numbers
IMCTOS, 4-2
IMSP, 2-4

W

Watchdog, 2-4, 5-2

X

XBIF Service
description, 1-2
Install XBIF Service command, 2-1
installing, 2-1, 2-4
X-Bus Interface Service, 1-2. *See also* XBIF Service.

UNISYS

Help Us To Help You

Publication Title

Form Number

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

Name Telephone number ( )

Title Company

Address

City State Zip code

Cut along dotted line

Tape

Please Do Not Staple

Tape

Fold Here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 817 DETROIT, MI

POSTAGE WILL BE PAID BY ADDRESSEE

UNISYS CORPORATION
PRODUCT INFORMATION
MS 18-007
2700 NORTH FIRST STREET
SAN JOSE, CA 95134-2028

43587260-000